Please amend the claims as follows.

1. **(Currently amended)** A method of detecting viral code in subject files, comprising:

creating an artificial memory region spanning one or more components of the operation system;

creating a custom version of an export table, wherein the custom version of the export table is associated with a plurality of entry points and wherein the entry points comprise predetermined values;

emulating execution of **at least a portion of** computer executable code in a subject file; ~~**and**~~

~~**detecting when the emulated computer executable code attempts to access the artificial memory region**~~

monitoring operating system calls by the emulated computer executable code;

identifying an operating system call that the emulated computer executable code attempted to access; and

deciding, based on the identified operating system call, whether the emulated computer executable code comprises viral code.

2. **(Canceled)**

3. **(Canceled)**

4. **(Currently amended)** The method of claim 1, further comprising:

~~**determining an operating system call that the emulated computer executable code attempted to access; and**~~

emulating functionality of the **identified** operating system call while monitoring the operating system call to determine whether the computer executable code is viral.

5. **(Currently amended)** The method of claim 1, further comprising:

monitoring accesses by the emulated computer executable code to the artificial memory region to detect looping **in the execution of the emulated computer executable code**; and

determining based on a detection of looping whether the emulated computer executable code is viral.

6. **(Canceled)**

7. **(Canceled)**

8. **(Original)** The method of claim 1, further comprising monitoring access by the emulated computer executable code to dynamically linked functions.

9. **(Previously presented)** The method of claim 8, wherein the artificial memory region spans a jump table containing pointers to the dynamically linked functions.

10. **(Currently amended)** A program storage device readable by a machine, tangible embodying a program of instructions executable by the machine to perform method steps for detecting viral code in subject files, the method steps comprising:

creating an artificial memory region spanning one or more components of the operating system;

creating a custom version of an export table, wherein the custom version of the export table is associated with a plurality of entry points and wherein the entry points comprise predetermined values;

emulating execution of **at least a portion of** computer executable code in a subject file; ~~**and**~~

~~**detecting when the emulated computer executable code attempts to access the artificial memory region**~~

monitoring operating system calls by the emulated computer executable code;

identifying an operating system call that the emulated computer executable code attempted to access; and

deciding, based on the identified operating system call, whether the emulated computer executable code comprises viral code.

11. **(Currently amended)** A computer system, comprising:

a processor; and

a program storage device readable by the computer systems, tangibly embodying a program of instructions executable by the processor to perform method steps for detecting viral code in subject files, the method comprising:

creating an artificial memory region spanning one or more components of the operating system;

creating a custom version of an export table, wherein the custom version of the export table is associated with a plurality of entry points and wherein the entry points comprise predetermined values;

emulating execution of **at least a portion of** computer executable code in a subject file; ~~**and**~~

~~**detecting when the emulated computer executable code attempts to access the artificial memory region**~~

monitoring operating system calls by the computer executable code;

identifying an operating system call that the computer executable code attempted to access;

deciding, based on emulation of the identified operating system call, whether the computer executable code comprises viral code.

12. **(Currently amended)** A computer data signal embodied in a transmission medium which embodies instructions executable by a computer for detecting in a subject file viral code that uses calls to an operating system, the signal comprising:

a first segment comprising CPU emulator code, wherein the CPU emulator code emulates execution of **at least a portion of** computer executable code in the subject file;

a second segment comprising memory manager code, wherein the memory manager code creates an artificial memory region spanning components of the operating system and creates a custom version of an export table, wherein the custom version of the export table is associated with a plurality of entry points and wherein the entry points comprise predetermined values; ~~and~~

a third segment comprising monitor code, wherein the monitor code ~~detects when the emulated computer executable code attempts to access the artificial memory region~~ **monitors operating system calls by the emulated computer executable code;**

a fourth segment comprising identifying code, wherein the identifying code identifies an operating system call that the emulated computer executable code attempted to access; and

a fifth segment comprising deciding code, wherein the deciding code decides, based on the identified operating system call, whether the emulated computer executable code comprises viral code.

13. **(Currently amended)** The computer data signal of claim 12, further comprising:

~~a fourth segment comprising auxiliary code, wherein the auxiliary code determines an operating system call that the emulated computer executable code attempted to access; and~~

a **sixth** ~~fifth~~ segment comprising analyzer code, wherein the analyzer code **emulates functionality of** ~~monitors~~ the **identified** operating system call to determine whether the computer executable code is viral~~, while emulating continues~~.

14. **(Currently amended)** An apparatus for detecting in a subject file viral code that uses calls to an operating system, comprising:

a CPU emulator;

a memory manager component that creates an artificial memory region spanning one or more components of the operating system and that creates a custom version of an export table, wherein the custom version of the export table is associated with a plurality of entry points and wherein the entry points comprise predetermined values; and

a monitor component, wherein the CPU emulator emulates execution of **at least a portion of** computer executable code in the subject file, and the monitor **component: ~~component detects when the emulated computer executable code attempts to access the artificial memory region~~**

monitors operating system calls by the emulated computer executable code;

identifies an operating system call that the emulated computer executable code attempted to access; and

decides, based on the identified operating system call, whether the emulated computer executable code comprises viral code.

15. **(Currently amended)** The apparatus of claim 14, further comprising:

an auxiliary component; and

an analyzer component,

wherein the auxiliary component **emulates functionalities of the identified ~~determines an~~** operating system call **~~that the emulated computer executable code attempted to access~~**, and the **monitor** **~~analyzer~~** component monitors the operating system call to determine whether the computer executable code is viral, while emulation continues.

16. **(Original)** The apparatus of claim 14, wherein the auxiliary component emulates functionalities of the operating system call.

17. **(Currently amended)** The apparatus of claim 14, wherein the analyzer component:

monitors accesses by the emulated computer executable code to the artificial memory region to detect looping **in the execution of the emulated computer executable code**; and

determines based on a detection of looping whether the emulated computer executable code is viral.

18. **(Canceled)**

19. **(Canceled)**

20. **(Original)** The apparatus of claim 14, wherein the artificial memory region created by the memory manager component spans a jump table containing pointers to dynamically linked functions, and the monitor component monitors access by the emulated computer executable code to the dynamically linked functions.